In the Claims:

Please amend the claims as follows:

1. (currently amended) A method to revalidate a compiler and a compiler execution environment intended for compilation of a user-written program for safety control in an industrial control system after use of the compiler and the compiler execution environment, the method comprising:

compiling a test program a first time with a compiler which test program is defined in a control language;

validating the compiler and the compiler execution environment by verifying that the test program executes correctly;

generating a first software element means derived from the compiled test program intended for later comparison purposes;

compiling the test program a second time after the compilation of a user-written program wherein the test program and the user-written program are compiled with the compiler used to compile the text program the first time;

generating a second software element means intended for a comparison based on the second compilation of the test program;

comparing the first software element means with the second software element means to find to determine whether errors were introduced between the first and the second compilation;

enabling, provided that the revalidation indicates no errors in the compiler and the compiler execution environment, the user-written program to execute in a device with safety

2

features for control of real world entities; and

when the user-written program is enabled, executing said user-written program in said

device with safety features for control of real world entities.

2. (previously amended) The method according to claim 1, wherein the comparing is

performed in the same workstation or general-purpose computer as that in which the compiler is

executing.

3. (currently amended) The method according to claim 1, wherein the software ~~element~~

~~means~~ is a check-sum or a code for cyclic redundancy check.

4. (previously amended) The method according to claim 3, wherein the comparing is

performed in the device with safety features.

5. (previously amended) The method according to claim 4, wherein the comparing

further comprises downloading a variable that changes over time, which is downloaded in the

same message as the check-sum or code to the device, where the variable that changes over time

is used to achieve a change in the message.

6. (previously amended) The method according to claim 1, wherein the test program is

defined in a control language derived from the standard IEC 6-1131.

7. (currently amended) A computer program product, comprising:

computer readable medium; and

computer program instructions recorded on the computer readable medium and executable by a processor for carrying out a method to revalidate a compiler and a compiler execution environment intended for compilation of a user-written program for safety control in an industrial control system after use of the compiler and the compiler execution environment, the method comprising:

compiling a test program a first time <u>with a compiler</u> which test program is defined in a control language;

validating the compiler and the compiler execution environment by verifying that the test program executes correctly;

generating a first software <u>element</u> ~~means~~ derived from the compiled test program intended for later comparison purposes;

compiling the test program a second time after the compilation of a user-written program <u>wherein the test program and the user-written program are compiled with the compiler used to compile the text program the first time</u>;

generating a second software <u>element</u> ~~means~~ intended for a comparison based on the second compilation of the test program;

comparing the first software <u>element</u> ~~means~~ with the second software <u>element</u> ~~means to find~~ <u>to determine whether</u> errors <u>were</u> introduced between the first and the second compilation;

enabling, provided that the revalidation indicates no errors in the compiler and the compiler execution environment, the user-written program to execute in a device with safety features for control of real world entities; and

when the user-written program is enabled, executing said user-written program in said

4

device with safety features for control of real world entities.

8. (cancelled)

9. (cancelled)